ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

VIDEO CONFERENCING USING REAL TIME TRANSMISSION PROTOCAL

Prof. Dr. K. P. Satheyamoorthy

Dean, Dept of EEE, **Dr**. M.G.R Educational and Research Institute [University], Chennai, Tamil Nadu, India

ABSTRACT: The goal of the project has been the design and implementation of Video Conferencing over the network. The Software has to not only guarantee better real time service, but also good performance with respect to the playback of the streamed media data. Thus it is required to create the software to allow the network within the permissible time. To do so, it is necessary to create a way package MPEG video into Real Time Protocol (RTP) packets. RTP is a best-effort protocol geared toward real time transmission and thus toward multimedia data, It is also necessary to provide a graphical interface at the client site to display the list of servers and media files available for playing and also a JMF based real-time player has to be proved with proper GUI for providing interactivity with the client. This included the control panel options such as forwarding, pausing volume control and closing the player.

Keywords: RTP, Videoconferencing, GUI, MPEG

I.INTRODUCTION

The proposed system is mainly divided into three modules.

- ✓ Client Interface
- ✓ Transmitting Module

Video Capture Module

- Session Allocation Video
- Data Transmission

Receiving Module

- Video Data Receiving
- Session Separation
- Video Execution

Client Interface**:**

This graphical module is provided at the client site and acts as simple browser. It displays the list of servers available with their IP-address and the media files corresponding to the Server chosen by client. When the client chooses the server and the media file, the transmitting module at the server site is invoked through a thread.

Transmitting Module:

This module transmits media over the network from the client which is provided at the server site. It receives three parameters namely the source locator, Client IP-address and a base destination port number for the tracks the Client interface module.

Receiving Module:

This module receives media streams which are transmitted from the server. This module is provided at the client site. It performs t he following tasks:

- ➢ Opens one RTP session per session address given.
- ➢ Listener for the New Receive Stream Event from the Receive Stream Listener.
- ➢ Creates a JMF player for the stream received for playback.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

II. BASIC REAL-TIME TRANSPORT PROTOCOL DETAILS

RTP is a transport protocol for real time applications. This protocol is used the transport of real time data, including audio and video. It can be used for media-on-demand as well as interactive services such as telephony. RTP provides support for applications with real time properties such as continuous media (e.g. audio and video), including loss detection security and content identification. UDP/IP is RTP's initial target networking environment.

RTP does not address resource reservation and does not guarantee quality-of-service for real-time services. The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality. RTP and RTCP are designed to be independent of the underlying transport and network layers. RTP itself does not provide any mechanism to ensure timely deliver or provide other quality of service guarantees, but relies on lower-layer services to do so. It doesn't guarantee delivery or prevent out-of order delivery, nor does it assume that underlying network is reliable and delivers packets in sequence. The sequence numbers included in RTP allow the receiver to reconstruct the sender's packet sequence, but sequence numbers might also be used to determine the proper location of a packet.

Streaming Media:

A key characteristic of time based media is that it requires timely delivery and processing. Once the flow of media data begins, there are strict timing deadlines that must be met, both in terms of receiving and presenting the data. For this reason, time based media is often referred to as streaming media and it is delivered in a steady stream that must be received and processed within a particular time to produce acceptable results.

For example, when a movie is played, if the media data cannot be delivered quickly enough, there might be odd pauses and delays in play back. On the other hand, if the data cannot be received and processed quickly enough, the movie might appear jumpy as data is lost.

Content type:

The format in which the media data is stored is referred to as its content type. Quick time, MPEG, and WAV are all examples of content types.

Media streams:

A media stream is the media data obtained from a local file, acquired over the network, or captured from camera or microphone. Media streams often contain multiple channels of data called tracks. Media streams that contain multiple tracks are often referred to as multiplexed or complex media streams. Demultiplexing is the process of extracting individual tracks from a complex media stream.

A track's type identifies the kind of data it contains, such as audio or video. The format of a track defines how the data for the track is structured. A media stream can be identified by its location and the protocol used to access it. For example, a URL might be used to describe the location of a Quick Time file on a local or remote system. A media locator provided a way to identify the location of a media stream when a URL can't be used.

Media Format details:

The following tables identify some of the characteristics of common media formats. When selecting a format, its important to take into account the characteristics of the intended audience. For example, as in this case we are delivering media content via the net-work, thus special attention should be paid to the net-work, thus special attention should be paid to the band width requirements. The CPU column characterizes presentation of the specified format. The bandwidth requirements column characterizes the transmission speed necessary to send or receive data quickly enough for optimal presentations.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

Format	Content type	Quality	CPU Requirements	Bandwidth Requirements
MP EG – 1	MPEG	High	High	High
H.26 1	AVI RTP	Low	Medium	Medium
H.26 3	Quick Time AVI RTP	Medium	Medium	Low
JPEG	Quick Time AVI RTP	High	High	High

Table1: Common Audio Formats

Format	Content type	Quality	CPU Requirements	Bandwidth Requirements
MP EG – 1	MPEG	High	High	High
PCM	AVI Quick Time WAV	High	Low	High
G.72	AVI Quick Time WAV	High	Low	High
3.1	RTP	Ium		

Table1: Common video Formats

Media Presentation Quality:

The quality of the presentation of a media steam depends on several factors, including:
1. The compression scheme used.
2. The processing capability of the playback system.
3. The bandwidth available (for media streams acquired over the network)To achieve high-quality video presentations, the number of frames displayed in each period of time (the frame rate) should be as high as possible.

III.TESTING

Test Plan:
This part solves the question “what to do before implementing the system which includes various testing of the system and the change over from the existing system to the proposed system is discussed in detail.

System Testing:

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

Several testing techniques are discussed below to be aware of any installation conflicts .The system testing is been done in several ways out of which some of them are discussed below.

Functional Testing:
The application SET in tested by filling the identity frame which consists of User-Id, password and ODBC connections in which the client password for which the browser is been selected .Once not connected the application is failed.

Stress Test:
In the client side only the authorized client can access the site to check the required details provided by the administration with limitations and the strength of the application software where places the important role.

Structural Test:
The System is also tested by other members of the SSPL team with exercising the internal logic of the application program I such a way that they switched from one application on to another operation and thus retaining the reliability of the application.

Change Over:
Change over is the process of converting one system to another. The new system may involve installing new software and preparing data. There are three methods of handlings a systems conversion.

Direct Change Over:
This method is complete replacement of the old system by proposed system in one move. It should be a bold move, which should be undertaken only when everyone in the firm has confidence on the proposed system. This method is potentially the least expensive but the most risky.

Parallel Running:
Parallel running or operation means processing current data by new system to cross check the results with the whole system. Its main attraction is that the old system is kept alive and operationally until the proposed system has been proved as success one. Its main disadvantage is extra cost.

Pilot Running:
Pilot running is similar to the concept of parallel running. This method is more like an extended system test, but is may be considered as a more practical form of change over for organization reasons.

Black Box Testing:
The Black box testing methods focus on the functional requirements of the software therefore black box testing enables the software engineer to drive sets of input condition as that will exercise all the functional requirements for an program. The black box testing is not an alternative to white box techniques, but it is a complementary approach that is likely to uncover a different class of errors than white box method the block box testing attempts to find errors in the following categories.

- Incorrect or missing functions
- Interface errors
- Errors in data structured or external database access
- Performance errors
- Initialization and termination errors

White Box Testing:
White box testing is a case design method that uses the control structure of the procedural design to derive test cases. Using the white box testing methods, the software engineer can drive test cases with the following qualities.

- The test case guarantee that all independent paths with in module have been exercise at least once.
- The test cases exercise all logical decisions on their true and false sides.
- The test cases execute all loops at their operational bounds and
- The test cases execute internal data structures to ensure their validity.

There are some software defects in the white box testing.

- Logic errors and incorrect assumptions are inversely proportional to the probability that a program path will be executed. Errors enter into our work, when we design and implement functions, conditions, or control that are out of the mainstream.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

- It is believed that a logical path is not likely to be executed when it may be executed on a regular basis. The assumptions about the flow of control and data may lead to make design errors that are uncovered only once when testing commences.

Typographical errors are random. When a program is translated in to programming language source code, it is likely that some typing errors will occur. Many will be uncovered by syntax checking mechanisms, but others will go undetected until testing begins.

IV. SYSTEM IMPLEMENTATION

This applicant is build around with the following modules:

1. Client interface
2. Transmitting module
3. Receiving module

Client Interface:
This graphical module is developed in Java using the code warrior and is provided at the client site and acts as a simple browser. It displays the list of servers available with their IP-address and the media files corresponding to the server chosen by the client.
When the client chooses the server site is involved through a thread. Three parameters are send to the transmitting module they are source locator, client IP-address and the base destination port number. After some delay another thread is created which invokes the Receiving module. The RTP session address are send to the Receiving module as its parameters .In the event of the JMF Player being closed, both the transmitting and Receiving module are stopped.

Transmitting Module:
This module is also developed in Java and transmits media over the network from the server to the client using JMF and RTP. This module is provided at the site.

It receives three parameters namely the source locator, client IP-address and a base destination port number for the tracks from the client interface module.
The source locator can be
1. A file name such as: file:/c/videos/los.mpeg"
2. An http locator such as
http://funvideos.com/spech.avi
3. Or a capture data source such as "javasound://8000"
The IP-address should be of the computer that receives the transmission.

The base port number can be any port number that is not in use by any other services on the computer. For example, one can use "22222". Make sure that it is an even number. The first media sure that it is an even number. The first media track will be transmitted from the base port number plus 2 and so on.In this module firstly a processor is created for the media locator.

Data Source as;
ds = javax.media.Manager.create DataSource (locator);
This Data Source is used to create the processor as follows;
Private Process processor = null;
Processor = javax.media.Manager.create Processor (ds);

Then the processor is made to be configured. After the processor is configured the tracks are obtained from the processor.
Track Control []tracks = processor.getTrackControls();

The output content descriptor is set to RAW_RTP.Next the video track of formats JPEG and H.263 are manipulated for real-time scaling and the audio track left alone without any changes. Next the processor is made to be realized so as to create an output DataSource for JPEG/RTP.
Private Data Source data Output=null;
Data Output=processor.getDataOutput();

The get Data Output method returns a Processor object's output as a Data Source. This Data Source can be used as the input to another Player or Processor or as the input to a data sink. In this case it is used as an input to the Player.

Next the RTP Manager API is used to create sessions for each media track. Now the tracks are steamed through the RTP sessions. The first track is streamed from the base port number and the next one from base port number+2.

Receiving Module:

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

This module is also developed in Java and received the RTP media streams which are
Transmitted from the server. This module is provided at the client site.
It performs the following tasks:

- Opens one RTP session per session address given.
- Listens for the New Receive Stream Event the Receive Stream Listener.
- Creates a JMF Player for the stream received for playback.

It receives a string of RTP sessions addresses as its parameters from the Client interface module. The RTP sessions address format is IP-address/port number.
The IP address specifies the address of the computer which transmits the media i-e server and the port number is same as that is used in the Transmitting module. Since Mpeg data file contains two tracks thus two RTP session address are required.
For e.g.: If the IP address of the server is 192.168.0.4 and the base port number used is 4000 then the two RTP session address are192.168.0.4/4000 for the video and 192.168.0.4/4002 for the audio.
In this module depending upon the number of sessions addresses passed by the Client interface modules the RTP sessions are parsed. The Sessions address is parsed to see if a IP-address port and ttl are specified.
At the client side the buffer length is set, although may be set to different values depending upon the need.

```
Buffer Control bc=
 (Buffer Control)mgrs[i].getControl("javax.media. control. Buffer Control")
if(be!=null)
bc.setBufferLength(350);
```

Here the buffer length is set to 350. The module waits data arrives. The Session Listener Interface receives notification of changes in the state of the new participant. The Receiver Stream Listener receives notification of changes in the state of an RTP stream that's being received, here it is used to check for any RTP Payload Change Event if so an error message is displayed since the module does not handle it.

Next the format of the RTP Stream is obtained as follows:

```
ReceiveStreamstream=evt.getReceiveStream();

Stream = ((NewReceiveStreamEvent)evt). getReceiveStream();

        DataSource ds =streams.getDataSource();
RTP Control ctl (RTPControl)ds.getControl ("javax.media.rtp.RTPControl");
```

Then a player is created by passing the data source to the Media Manager

```
                Player p = javax.media.Manager. create Player(ds);
        If(p = =null)
return:
```

The Controller Listener is added to the player so as to handle the user invoked events. The player is then realized.

```
                p.addControlListener(this);
                p.realize();
```

After the player is realized it is started to playback the streams.

```
                p.star();
```

The player generally has two types of user interface components, a visual component and a Control-panel component. These components are obtained only after the player is realized.

Displaying a Visual Component:

A visual component is where a player presents the visual representation of its media. Even an audio player might have a visual component, Such as a wave form display or animated character.

To display a Player object's visual component, you:

1. Get the component by calling get visual Component.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

2. Add it to the applet's presentation space or application window.

The layout of the player components is controlled through the AWT layout manager.

```
if ((vc = p.getVisualCompomnent())!= null)
add("Center", vc);
```

Displaying a Control Panel Component

The Player has a control panel that allows the to control the media presentation. The Player might be associated with a set of buttons to start, stop and pause the media stream, and with a slider control to adjust the volume.

```
If((cc =p.getControlPanelComponent())!=null)
        Add("south",cc);
```

Impementing the Controller Listener Interface

To implement the Controller interface, you need to:

1. Implement the Controller Listener interface in a class.

2. Register that class as a listener by calling add Controller Listener on the Controller that you want to receive events from.

When a Controller posts an event, it calls controller Update on each registered listener.

Typically, controller Update is implemented as a series of if-else statements which responds to player events

Implementing controller Update.

```
If(event instance of Event Type){
…….
} else if (even instance of Other Even Type){
…….
}
```

Due to the limitations of JMF implementation, it is important to note that audio and video are not in tight synchronization.

V. RESULT AND DISCUSSION

The application was build around three modules, they are Client interface module, Transmitting module and Receiving module. The three modules were tested separately i-e the Client interface module was tested for its purpose which is to display the list of server IP addresses available and the corresponding media data available. The bugs encountered in this process were removed successfully. The Transmitting module was tested for its purpose of transmitting the media data from the server to the client through RTP streams. All the exceptions encountered were handled properly. Similarly the Receiving module was tested for its purpose of receiving the RTP streams and to play it using the JMF player. In this process also all the bugs and exceptions encountered were handled .These modules were later integrated and tested, necessary changes were made for the application so as to provide better performance. Although tight synchronization of the audio and video could not be achieved.

VI.CONCLUSION

The basic goal of this application was to stream the media data from the server to the client i-e to packetize the media data into RTP packets and then stream these packets through the RTP sessions and later to receive these streams and play them by creating the player. The goal was accomplished by providing a graphical interface module client site which displays the list of servers and media files available for playing. A transmitting module was developed at the server site to transmit the media data through RTP streams. A Receiving module was developed at the client site which receives the RTP streams and plays the streamed media. JMF based real-time player was also provided with proper GUI for providing interactivity with the client. The modules were tested and integrated to perform the task of Video Conferencing. The integration of the modules was successful providing the expected results. The process of acquisition of the media, its transmission, its reception and presentation was supported by the usage of JMF and RTP. RTP was

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

since it real-time applications. It is also used for the transport of real-time data, including audio and video and also for media-on-demand services .A major drawback was that of not being to provide tight synchronization of both audio and video owing to the limitations of the JMF implementation.

REFERENCES

[1] D.Hoffman, G.Fernando, V.Goyal and M.Civanlar, "RTP: A Transport Protocol for Real-Time Applications, "RFC 1889, January, 1996.
[2] D.Hoffman, G.Fernando, V.Goyal and M.Civanlar, "RTP payload format for MPEG1/MPEG2 video," RFC 2250, Internet Engineering Task Force, January, 1998.
[3] Tony Brey and Paul Chauffer : "Java Complete Reference"
[4] John Rovalds Steve raid and Tony Brey : "Video Streaming and Application"
[5] JavaMediaFramework: www.java.sun.com/jmf/apiodc.html
[6] RTP Resource: http://www.cs.columbia.edu/~hgs/rtp/
[7] RTSP Resource: http://www.cs.columbia.edu/~hgs/rtsp/